UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/692,770	10/24/2003	Christian Zander	7469 US	8945

30078 7590 03/31/2008

MATTHEW D. RABDAU
TEKTRONIX, INC.
14150 S.W. KARL BRAUN DRIVE
P.O. BOX 500 (50-LAW)
BEAVERTON, OR 97077-0001

EXAMINER	
FEARER, MARK D	
ART UNIT	**PAPER NUMBER**
2143	

MAIL DATE	DELIVERY MODE
03/31/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/692,770	ZANDER, CHRISTIAN	
	Examiner	Art Unit	
	MARK D. FEARER	2143	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *24 October 2003*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-16* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-16* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *24 October 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

- Applicant's Amendment of 20 December 2007 is acknowledged.
- Claim 17 is amended and renumbered as claim 16. Claim 17 is cancelled.
- Claims 1-16 are pending in the present application.
- This action is **FINAL**.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

The factual inquiries set forth in Graham v. John Deere Co., 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

Claims 1, 11-15 and 16 are rejected under 35 U.S.C. 103(a) as being obvious over Ehrhardt et al. (US A1) in view of Sierer et al. (US 20030036873 A1).

Consider claims 1 and 16. Ehrhardt et al. discloses a method of setting up a procedure of a communication taking place between instances which is executable on a

protocol tester of the type including the steps of selecting the instances involved in the communication, selecting a protocol layer on the basis of which the communication between the selected instances is to take place, selecting abstract communication interfaces of the protocol layer which are involved in the communication, selecting communication data, setting up a communication procedure executable between the instances through the protocol tester based on the several selecting steps, with the communication data selecting step being made graphically, with the parameters so selectable being allocated description files which are used in the setting up step ((“Accordingly the present invention provides a method of setting up a communication procedure between instances, one of which is a protocol tester, by executing the following steps on the protocol tester: selecting instances that are to take part in the communication procedure; selecting a protocol layer on the basis of the communication procedure; selecting abstract communication interfaces of the protocol layer for the communication procedure; selecting communication data; and automatically setting up through the protocol tester on the basis of the above selections the communication procedure. The selections at any one of the steps may be made graphically with parameters selected being assigned description files that are used in the setting up step.”) paragraph 0005). However, Ehrhardt et al. fails to disclose the ability to select from a plurality of functions, each having a graphical representation and a description file containing source code, or generating a configuration file at set-up time. Sierer et al. discloses the possibility of selecting from a plurality of functionalities ((“For example, a measurement device may have a programmable hardware element that is configurable

using the program(s). The user may choose to deploy the hardware configuration program(s) to various devices, e.g., using a GUI based configuration diagram.") paragraph 0044), each functionality being allocated a graphic representation and a description file (("In one embodiment a description file may be generated which identifies resources and features the task requires or that the user has selected. From this description file, G code (graphical code, e.g., National Instruments G graphical programming language) may be generated.") paragraph 0226), further comprising the steps of: creating a configuration file which is read in and interpreted at the time of the setting-up the communication procedure between the instances and from which is generatable at compile time an associated code (("a self-executing program file which operates to configure one or both of the client computer system and the one or more measurement devices with the configuration information.") paragraph 0031); and entering information into the configuration file including a call name of an additional functionality in the executable code (("In yet another embodiment, the measurement task specifier may be an API through which the user makes calls to generate the task specification. The measurement task specifier may thus generate the measurement task specification in response to user input.") paragraph 0167), a display form correlating with the additional functionality on a display on which it may be selected (("Thus, by performing the method described above, a user may specify a task, such as over a network, a server may generate a graphical program from the user-specified task specification and send the graphical program to the client, then the client may, if necessary, convert the graphical program to an executable form. The executable

program may then be run by the measurement system to perform the specified task.") paragraph 0273), and information on the description file which contains the executable source code of the additional functionality (("HDL code may then be generated from the G code (or directly from the description file), and eventually a program binary file, i.e., a hardware configuration program, for the FPGA generated from the HDL code. In these approaches, caching schemes may be used so that the number of compilations may be minimized.") paragraph 0226).

Therefore, it would have been obvious to a person skilled in the art at the time the invention was made to incorporate the ability to select from a plurality of functions, each having a graphical representation and a description file containing source code, or generating a configuration file at set-up time as taught by Sierer et al. with a method of setting up a communication procedure between instances, one of which is a protocol tester, by executing the following steps on the protocol tester: selecting instances that are to take part in the communication procedure; selecting a protocol layer on the basis of the communication procedure; selecting abstract communication interfaces of the protocol layer for the communication procedure; selecting communication data; and automatically setting up through the protocol tester on the basis of the above selections the communication procedure. The selections at any one of the steps may be made graphically with parameters selected being assigned description files that are used in the setting up step as taught by Ehrhardt et al. for the purpose of functionality and interoperability testing of protocol layers with flexible configuration and programming tools that allow test case to be defined at an abstract level.

Consider claim 11, and as applied to claim 1 above. Ehrhardt et al., as modified by Sierer et al., discloses a method wherein instances that are involved in a communication are graphically selected, a protocol layer is graphically selected and/or abstract communication interfaces of a protocol layer are graphically selected ((“… selecting instances that are to take part in the communication procedure; selecting a protocol layer on the basis of the communication procedure; selecting abstract communication interfaces of the protocol layer for the communication procedure; selecting communication data; and automatically setting up through the protocol tester on the basis of the above selections the communication procedure. The selections at any one of the steps may be made graphically with parameters selected being assigned description files that are used in the setting up step.“) paragraph 0005).

Consider claim 12, and as applied to claim 1 above. Ehrhardt et al., as modified by Sierer et al., discloses a method wherein abstract communication interfaces comprise SAPs (Service Access Points) ((“FIG. 3 shows the GUI of FIG. 2 in a different presentation mode, in this case for selecting a Service Access Point (SAP), as shown in field 32a. In field 32b there are further SAPs from which to choose. All SAPs shown in field 32b are offered for the selected emulation "isdn12."”) paragraph 0016).

Consider claim 13, and as applied to claim 1 above. Ehrhardt et al., as modified by Sierer et al., discloses a method wherein communication data is selected from a group consisting of PDUs (Protocol Data Units) and ASPs (Abstract Service Primitives) ((“FIG. 4 shows another presentation form of the GUI 10 of FIG. 2, with a format for the communication data (Abstract Service Primitives—ASPs, Protocol Data Units—PDUs)

now being used in a field 34 having so-called Message Pools.") paragraph 0017 ("FIG. 7 shows an isdn -PDU "SETUP.sub.--1" being incorporated into a flow diagram prepared graphically as a send message. ASPs with PDUs from the Message Pool selected earlier are offered in an entry mask 50. The PDU selected may be entered into a visually highlighted field 52. In a field 54 the user is offered further information on the ASP or PDU selected.") paragraph 0021).

Consider claim 14, and as applied to claim 1 above. Ehrhardt et al., as modified by Sierer et al., discloses a method wherein communication data selecting step comprises the steps of: d1) graphically selecting a data format and; d2) graphically setting up a communication sequence between the instances involved (("FIG. 1 shows a graphical user interface (GUI) 10 that allows in a first step graphically selecting instances taking part in a communication procedure. Graphical selection in connection means that a symbol or a text proposal is shown graphically on the GUI, such as on a personal computer (PC) screen, and may be selected by simple activation, i.e., by clicking on it with a "mouse." One of the instances is a protocol tester on which the method as described herein is made available, with the protocol tester in the present case emulating a component, TC.sub.--1. Using two buttons, "Add" 12 and "Delete" 14, a user may add further instances or delete instances listed. In a field 16 the compilation of instances is listed, while in another field 18 the compilation is shown as a diagram. In another field 19 the name of the instance may be selected, and in a further field 20 the instance type is shown. Two buttons, "Back" 22 and "Next" 24, allow the user to move from one level of the definition of the communication procedure to the next, both in the

direction of more detailed specifications and in the direction of higher-level presentations. A "Cancel" button 26 allows leaving a level, meaning that the changes made are reset. A "Help" button 28 offers the user further support.") paragraph 0014).

Consider claim 15, and as applied to claim 14 above. Ehrhardt et al., as modified by Sierer et al., discloses a method wherein source code is enterable (("For a test it is, therefore, necessary on the one hand to prepare graphical and textual documentation and on the other hand a source code or binary code that may be executed.") paragraph 0002 ("The method as recited in claim 6 wherein the graphically setting up step comprises the step of entering source code.") claim 7 ("The protocol tester as recited in claim 11 further comprising means for entering source codes.") claim 12).

Claims 2-3, and 6 are rejected under 35 U.S.C. 103(a) as being obvious over Ehrhardt et al. (US 20010015732 A1) as modified by Sierer et al. (US 20030036873 A1) and in further view of Kegel et al. (US 20030163519 A1).

Consider claims 2 and 6, and in view of claim 1 above. Ehrhardt et al., as modified by Sierer et al., discloses setting up a procedure of a communication taking place between instances executable on a protocol tester, selecting abstract communication interfaces of the protocol layer which are involved in the communication, selecting communication data, setting up a communication procedure executable between the instances through the protocol tester, and creating a configuration file which is read in and interpreted at the time of the setting-up the communication procedure between the instances and from which is generatable at compile time an

associated code. However, Ehrhardt et al., as modified by Sierer et al., fails to disclose a method wherein for each parameter of the additional functionality to be entered and for the result of the additional functionality is also entered into the configuration file a name and a type. Kegel et al. discloses HTML plugins with dynamic configuration files that specify such things as names, types, tags and links ((“4TABLE 1 !!Author!! Inserts text representing details of the web (takes no parameters) site's author, as optionally specified in an .ini configuration file. !!ChildLinks <link> <dir>!! Inserts HTML code representing a list of <link> This specifies the HTML links to the menu (node) URL for each formatting which should sub-folder leading from the current folder be used with the link plain creates a standard textual hyperlink. image creates an image-based hyperlink, using an image of the same name as the destination folder. table creates cells within a table containing one textual hyperlink in each cell (designed to easily create button bars). The table itself should be defined manually in the template file. tablewithimages creates a table as above but uses image- base hyperlinks. . . . any other entry specifies a link template which can contain any other desired HTML formatting. <dir> This specifies the direction in which a list of links should run. This parameter will have no effect if there is only a single link to create. horiz includes formatting to create a horizontal list vert includes formatting to create a vertical list !!Comment!! Inserts the full textual content of one or (takes no parameters) more comment files. If used within a node template, this plugin will insert the contents of all files with the .cmt extension which do not have the same names as document files. If used within a document template, this plugin will insert the contents of the single .cmt file with the same name as

that document. The plugin will add HTML paragraph formatting to the text content unless HTML formatting tags are already present. !!CousinLinks <link> <dir> <exclude>!! Inserts HTML code representing a list of <link>") Table I pages 8-11).

Therefore, it would have been obvious to a person skilled in the art at the time the invention was made to incorporate HTML plugins with dynamic configuration files as taught by Kegel et al. with setting up a procedure of a communication taking place between instances executable on a protocol tester, selecting abstract communication interfaces of the protocol layer which are involved in the communication, selecting communication data, setting up a communication procedure executable between the instances through the protocol tester, and creating a configuration file which is read in and interpreted at the time of the setting-up the communication procedure between the instances and from which is generatable at compile time an associated code as taught by Ehrhardt et al., as modified by Sierer et al., for the purpose of dynamic configuration.

Consider claim 3, and in view of claim 2 above. Ehrhardt et al., as modified by Sierer et al., discloses setting up a procedure of a communication taking place between instances executable on a protocol tester, and creating a dynamic configuration file. However, Ehrhardt et al., as modified by Sierer et al., fails to disclose a method wherein the display form is selected from the group consisting of a graphic symbol for the additional functionality, the graphic symbol being allocated a graphic file in which the graphic symbol of the additional functionality is implemented. Kegel et al. discloses a user configurable template file ((“A process for handling graphical file references will now be described with reference to FIG. 12. If the template contains graphic file

references, i.e. references to files with the suffix .gif or the like, a search is carried out for them at step S2.14. The search is initially carried out in the current folder and if not located, the search moves up one level in the hierarchical folder structure. This search process is repeatedly carried out at higher levels until the corresponding file is found, as illustrated by step S2.15. At step S2.16, the graphic is inserted into the web page. For example, considering the Home page shown in FIG. 7, the HTML template lemon.tpl given above includes reference to "Logo.gif". This file is included in the "Lemon Source" folder 33 and results in the image 54 being included in the Home Page shown in FIG. 7. If any of the node templates refer to "Logo.gif" then the file is located and inserted into the corresponding HTML for the node according to the process shown in FIG. 12. Also the plugin !!DocumentImage!! can be used to import a graphical file into HTML produced by a template. When used in a within a node template, this plugin will insert a reference to an image which has the same name as the current folder. By way of example, referring to FIG. 6, the plugin !!DocumentImage!! is used in the template node.tpl with the result that the file "meals.gif" 48 which has the same name as the folder Meals 36 produces a corresponding image 48 in the displayed web page 49 shown in FIG. 6. When used within a document template, for example document.tpl above, the plugin !!DocumentImage!! will insert a reference to an image which has the same name as the current document.") paragraphs 0095-0096).

Therefore, it would have been obvious to a person skilled in the art at the time the invention was made to incorporate user configurable template files as taught by Kegel et al. with setting up a procedure of a communication taking place between

instances executable on a protocol tester as taught by Ehrhardt et al., as modified by Sierer et al., for the purpose of creating interface menus.

Claim 4 is rejected under 35 U.S.C. 103(a) as being obvious over Ehrhardt et al. (US 20010015732 A1) as modified by Sierer et al. (US 20030036873 A1) and in further view of Kouznetsov et al. (US 6931546 B1).

Consider claim 4, and as applied to claim 1 above. Ehrhardt et al., as modified by Sierer et al., discloses setting up a procedure of a communication taking place between instances executable on a protocol tester. However, Ehrhardt et al., as modified by Sierer et al., fails to disclose a method wherein the description file and/or the executable code are formulated in Forth, Jscript or VBScript. Kouznetsov et al. discloses an appliance interface that incorporates markup language (("Appliance 117 preferably includes a scripting interface 203 for executing script, including scripts 213, provided by server 211. Example scripting interfaces include VBScript, Jscript, JavaScript. Markup language documents such as extensible markup language (XML) is an alternative tool enabling client-side program execution. Collectively, these are tools that enable execution of code on a client machine (e.g., appliance 117) that is generated by server 211.") column 8 lines 45-52).

Therefore, it would have been obvious to a person skilled in the art at the time the invention was made to incorporate scripting interfaces including VBScript, Jscript, JavaScript as taught by Kouznetsov et al. with setting up a procedure of a communication taking place between instances executable on a protocol tester as

taught by Ehrhardt et al., as modified by Sierer et al., for the purpose of simulation, test, analysis, and debugging of communications interfaces.

Claims 5 is rejected under 35 U.S.C. 103(a) as being obvious over Ehrhardt et al. (US 20010015732 A1) as modified by Sierer et al. (US 20030036873 A1) and in further view of Burkhardt et al. (US 20030037326 A1).

Consider claim 5, and as applied to claim 1 above. Ehrhardt et al., as modified by Sierer et al., disclose a procedure of a communication taking place between instances executable on a protocol tester. However, Ehrhardt et al., as modified by Sierer et al., fails to disclose a method wherein a configuration file is implemented as a text file selected from a group consisting of an INI format and an XML format. Burkhardt et al. discloses a text configuration file in .INI or XML format ((“According to a preferred embodiment of the invention, computer 110 operates in an environment in which scripts are used to create the reference system image. A configuration script (e.g., a text file such as WINBOM.INI) preferably directs the installation utility FACTORY.EXE and controls the order in which programs are installed or staged on the reference system and re-boots the reference system as necessary. In this manner, staged system hard drive 170 is established for use in pre-configuring one or more target or destination machines. It is to be understood that the script need not be in the form of a text file in a .INI file format but may be, for example, an XML file or even a binary format.”) paragraph 0027).

Therefore, it would have been obvious to a person skilled in the art at the time the invention was made to incorporate a text configuration file in .INI or XML format as taught by Burkhardt et al. with setting up a procedure of a communication taking place between instances executable on a protocol tester as taught by Ehrhardt et al., as modified by Sierer et al., for the purpose of defining data formats.

Claim 7 is rejected under 35 U.S.C. 103(a) as being obvious over Ehrhardt et al. (US 20010015732 A1) as modified by Sierer et al. (US 20030036873 A1) and in further view of Ames et al. (US 6151567 A).

Consider claim 7, and as applied to claim 6 above. Ehrhardt et al., as modified by Sierer et al., disclose a procedure of a communication taking place between instances executable on a protocol tester. However, Ehrhardt et al., as modified by Sierer et al., fails to disclose a method wherein the configuration file further includes information on how many functionalities are stored in it. Ames et al. discloses a configuration file that defines parameters and modes ((“The server process 140 calls the device driver 154 to configure the ARINC 629 cards 114 and 118 (see FIG. 2) installed in the computer 102 (see FIG. 2) based on the parameters and modes defined in the configuration file 176 and loaded into the database 174.”) column 8 lines 9-13).

Therefore, it would have been obvious to a person skilled in the art at the time the invention was made to incorporate a configuration file that defines parameters and modes as taught by Ames et al. with setting up a procedure of a communication taking

place between instances executable on a protocol tester as taught by Ehrhardt et al., as modified by Sierer et al., for the purpose of parameter tuning.

Claims 8-9 are rejected under 35 U.S.C. 103(a) as being obvious over Ehrhardt et al. (US 20010015732 A1) as modified by Sierer et al. (US 20030036873 A1) and in further view of Tillman et al. (US 20020103937 A1).

Consider claim 8, and as applied to claim 1 above.8. Ehrhardt et al., as modified by Sierer et al., disclose a procedure of a communication taking place between instances executable on a protocol tester. However, Ehrhardt et al., as modified by Sierer et al., fails to disclose a method wherein for implementation into executable source code of additional functionality, a call of additional functionality is entered together with its call name. Tillman et al. discloses named routines that call for source code input ((“Executable software 340 is generated from protocol specification 140 in several steps. First, a specification compiler 310 accepts protocol specification 140 and generates packet decoder source code 320. For instance, specification compiler 310 accepts protocol specification 140 in a syntax described below in Section Error! Reference source not found and produces packet decoder source code in the syntax of the C++programming language. In addition to protocol specification 140, message processing source code 325, for instance also specified in the syntax of the C++ programming language, defines how individual messages generated by packet decoder 110 will be processed. Finally, communication processor source code 315 includes a specification of overall routines to be executed by the communication processing device, for example, including input routines to accept packet sequence 115 and

routines to invoke the routines defined in packet decoder source code 330.") paragraph 0041 ("… for a call of the subroutine named ActionSubr with its argument being the value of the fielda field.") paragraph 0076).

Therefore, it would have been obvious to a person skilled in the art at the time the invention was made to incorporate named routines that call for source code input as taught by Tillman et al. with setting up a procedure of a communication taking place between instances executable on a protocol tester as taught by Ehrhardt et al., as modified by Sierer et al., for the purpose of protocol compiling and low level RPC programming.

Consider claim 9, and as applied to claim 8 above. Ehrhardt et al., as modified by Sierer et al., disclose a procedure of a communication taking place between instances executable on a protocol tester. However, Ehrhardt et al., as modified by Sierer et al., fails to disclose a method wherein prior to a function call, parameters required by additional functionality are handed over, and after the call the result of the additional functionality is handed over. Tillman et al. discloses a function wherein parameters are passed (("The definition portion of a field statement can be used to associate values, such as numeric constants, with symbolic names that are passed as the parameters of the messages corresponding to that field. The definition portion of the statement is also used to define the action the packet decoder should take when it encounters this field.") paragraph 0060).

Therefore, it would have been obvious to a person skilled in the art at the time the invention was made to incorporate a function wherein parameters are passed as taught by Tillman et al. with setting up a procedure of a communication taking place between instances executable on a protocol tester as taught by Ehrhardt et al., as modified by Sierer et al., for the purpose of test methods incorporating scripting language.

Claim 10 is rejected under 35 U.S.C. 103(a) as being obvious over Ehrhardt et al. (US 20010015732 A1) as modified by Sierer et al. (US 20030036873 A1) and in further view of Underwood (US 6523027 B1).

Consider claim 10, and as applied to claim 1 above. Ehrhardt et al., as modified by Sierer et al., disclose a procedure of a communication taking place between instances executable on a protocol tester. However, Ehrhardt et al., as modified by Sierer et al., fails to disclose a method wherein reading-in of a description file occurs via an include command. Underwood discloses Server Side Include statements to allow for quick and efficient modification of multiple web pages (("Server side includes allow HTML authors to place commands inside their portion of the present descriptions that cause output to be modified whenever that portion of the present description is accessed by a user.") column 110 lines 61-65).

Therefore, it would have been obvious to a person skilled in the art at the time the invention was made to incorporate Server Side Include statements as taught by Underwood with setting up a procedure of a communication taking place between

instances executable on a protocol tester as taught by Ehrhardt et al., as modified by Sierer et al., for the purpose of efficient modification of multiple web pages.

Response to Arguments

In response to Applicant's Arguments filed 20 December 2007; Applicant asserts that Ehrhardt et al. reference, US 20010015732 A1 is commonly owned with Applicant's invention and therefore may be disqualified under 35 U.S.C. 103(c). Examiner respectfully disagrees. The Ehrhardt et al. reference, US 20010015732 A1, though commonly owned with assignee of Applicant's invention, was published as a US Patent Application more than one year prior to Applicant's invention, thus qualifying as a 35 U.S.C. 102(b).

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of

the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any response to this Office Action should be faxed to (571) 273-8300 or mailed to:

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Hand-delivered responses should be brought to

Customer Service Window
Randolph Building
401 Dulany Street
Alexandria, VA 22314

Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Mark Fearer whose telephone number is (571) 270-1770. The Examiner can normally be reached on Monday-Thursday from 7:30am to 5:00pm.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Nathan Flynn can be reached on (571) 272-1915. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status

information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free) or 571-272-4100.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist/customer service whose telephone number is (571) 272-2600.

Mark Fearer
M.D.F./mdf
March 25, 2008

/Kenny S Lin/
Primary Examiner, Art Unit 2152